Automatic Soccer Video Analysis and Summarization

Article *in* IEEE Transactions on Image Processing · February 2003

DOI: 10.1109/TIP.2003.812758 · Source: PubMed

3 authors, including:

Ahmet Ekin
Philips
67 PUBLICATIONS **2,219** CITATIONS
SEE PROFILE

A. Murat Tekalp
Koc University
534 PUBLICATIONS **14,533** CITATIONS
SEE PROFILE

Automatic Soccer Video Analysis and Summarization

Ahmet Ekin, A. Murat Tekalp, *Fellow, IEEE*, and Rajiv Mehrotra

***Abstract*—We propose a fully automatic and computationally efficient framework for analysis and summarization of soccer videos using cinematic and object-based features. The proposed framework includes some novel low-level soccer video processing algorithms, such as *dominant color region detection, robust shot boundary detection, and shot classification,* as well as some higher-level algorithms for *goal detection, referee detection,* and *penalty-box detection*. The system can output three types of summaries: i) all slow-motion segments in a game, ii) all goals in a game, and iii) slow-motion segments classified according to object-based features. The first two types of summaries are based on cinematic features only for speedy processing, while the summaries of the last type contain higher-level semantics. The proposed framework is efficient, effective, and robust for soccer video processing. It is *efficient* in the sense that there is no need to compute object-based features when cinematic features are sufficient for the detection of certain events, e.g., goals in soccer. It is *effective* in the sense that the framework can also employ *object-based features* when needed to increase accuracy (at the expense of more computation). The efficiency, effectiveness, and the robustness of the proposed framework are demonstrated over a large data set, consisting of more than 13 hours of soccer video, captured at different countries and conditions.**

***Index Terms*—Cinematic features, object-based features, semantic event detection, shot classification, slow-motion replay detection, soccer video processing, soccer video summarization.**

I. Introduction

SPORTS video distribution over various networks should contribute to quick adoption and widespread usage of multimedia services worldwide, because sports video appeals to large audiences. Processing of sports video, for example detection of important events and creation of summaries, makes it possible to deliver sports video also over narrow band networks, such as the Internet and wireless, since the valuable semantics generally occupy only a small portion of the whole content. The value of sports video, however, drops significantly after a relatively short period of time [1]. Therefore, sports video processing needs to be completed *automatically*, due to, otherwise, its intimidating size, in *real*, or *near real-time*, and the processing results must be *semantically meaningful*. In this paper, we propose a novel soccer video processing framework that satisfies these requirements.

Semantic analysis of sports video generally involves use of *cinematic* and *object-based* features. Cinematic features refer to those that result from common video composition and production rules, such as shot types and replays.[1] Objects are described by their spatial, e.g., color, texture, and shape, and spatio-temporal features, such as object motions and interactions [2]. Object-based features enable *high-level domain analysis*, but their extraction may be *computationally costly* for real-time implementation. Cinematic features, on the other hand, offer *a good tradeoff* between the computational requirements and the resulting semantics.

In the literature, object color and texture features are employed to generate highlights [3] and to parse TV soccer programs [4]. Object motion trajectories and interactions are used for football play classification [5] and for soccer event detection [6]. Both [5] and [6], however, rely on pre-extracted accurate object trajectories, which were obtained manually in [5]; hence, they are not practical for real-time applications. LucentVision [7] and ESPN K-Zone [8] track only specific objects for tennis and baseball, respectively. The former analyzes trajectory statistics of two tennis players and the ball. The latter tracks the ball during pitches to show, as replays, if the strike and ball decisions are correct. The *real-time* tracking in both systems is achieved by extensive use of *a priori* knowledge about the system setup, such as camera locations and their coverage. Therefore, their application to TV broadcast soccer video, which is the focus of this paper, is limited. Cinematic descriptors are also commonly employed. The plays and breaks in soccer games are detected by frame view types in [9] and by motion and color features in [10]. Li and Sezan summarize football video by play/break and slow-motion replay detection using both cinematic and object descriptors [11]. Scene cuts and camera motion parameters are used for soccer event detection in [12] where usage of very few cinematic features prevents reliable detection of multiple events. Similarly, camera motion and some object-based features are employed in [13] to detect certain events in soccer video. However, unlike a fully automatic system proposed in this paper, object-based features are manually annotated in [13]. A mixture of cinematic and object descriptors is employed in [14] and [15]. Motion activity features are proposed for golf event detection [16]. *Text* information from closed captions and visual features are integrated in [17] for event-based football video indexing. *Audio* features, alone, are proposed to detect hits and generate baseball highlights [18]. In previous works, such as [11], [16], [18], the summaries have been generated by concatenating a pre-defined temporal interval about the set of keyframes that sat-

Manuscript received July 19, 2002; revised February 12, 2003. This work was supported in part by the National Science Foundation under Grant IIS-9820721 and Eastman Kodak Company. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Bruno Carpentieri.

A. Ekin is with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627-0126 USA (e-mail: ekin@ece.rochester.edu).

A. M. Tekalp is with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627-0126 and also with the College of Engineering, Koc University, Istanbul, Turkey (e-mail: tekalp@ece.rochester.edu; mtekalp@ku.edu.tr).

R. Mehrotra is with the Entertainment Imaging Division, Eastman Kodak Company, Rochester, NY 14650 (e-mail: rajiv.mehrotra@kodak.com).

Digital Object Identifier 10.1109/TIP.2003.812758

[1]Camera motion, when used within a cinematic context, is regarded as a cinematic feature.

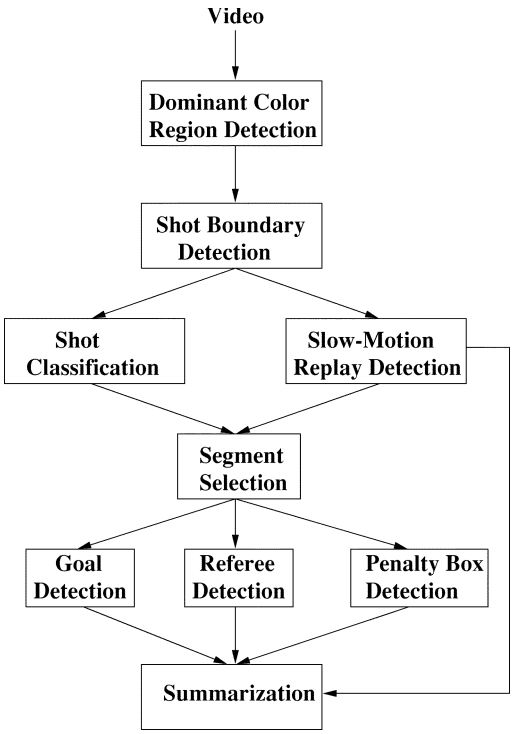


Fig. 1. Flowchart of the system.

isfy the saliency of the selected features, such as motion activity in [16] and audio in [18]. In contrast, we provide key clip summaries with adaptive durations to better capture the semantic events.

In this paper, we propose a new framework for *automatic*, *real-time* soccer video analysis and summarization by systematically using cinematic and object features. A flowchart of the proposed framework is shown in Fig. 1. The main contributions are as follows.

1) We propose new dominant color region and shot boundary detection algorithms that are *robust to variations in the dominant color*. The color of the grass field may vary from stadium to stadium, and also as a function of the time of the day in the same stadium. Such variations are automatically captured at the initial training stage of our proposed dominant color region detection algorithm. Variations during the game, due to shadows and/or lighting conditions, are also compensated by automatic adaptation to local statistics.

2) We propose two novel features for shot classification in soccer video. They provide *robustness to variations in cinematic features*, which is due to slightly different cinematic styles used by different production crews. The proposed shot classification algorithm provides as high as 17.5% improvement over an existing algorithm as shown in Section V.

3) We introduce new algorithms for automatic detection of i) goal events, ii) referee, and iii) penalty box in soccer videos. Goals are detected based solely on cinematic features resulting from common rules that are employed by the producers after goal events to provide a better visual experience for TV audiences. Distinguishing jersey color of the referee is used for fast and robust referee detection. Penalty box detection is based on the *three-parallel-line* rule that uniquely specifies the penalty box area in a soccer field.

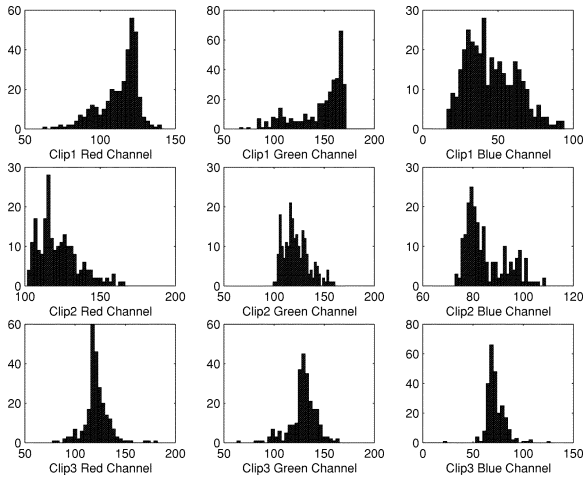


Fig. 2. Grass color histograms collected in 45-minute segments demonstrate the variations: Clip1 and Clip2 are the games in the same stadium, under spotlight and daylight conditions, respectively, Clip3 is a game in a different stadium under spotlights. RGB mean grass values are (113,146,46), (122,122,85), and (121,129,72), while standard deviations are (13.2,24.4,16.7), (13.6,12.7,8.3), and (12.4,13.9,9.4), respectively.

4) Finally, we propose an efficient and effective framework for soccer video analysis and summarization that combines these algorithms in a scalable fashion. It is *efficient* in the sense that there is no need to compute object-based features when cinematic features are sufficient for the detection of certain events, e.g., goals in soccer. It is *effective* in the sense that the framework can utilize *object-based features* when needed to provide more detailed summaries (at the expense of more computation). Hence, the proposed framework is adaptive to the requirements of the desired processing.

We describe the proposed low-level algorithms for dominant color region detection, shot boundary detection, shot classification, and slow-motion replay detection in the next section. Section III presents proposed higher-level methods for goal detection, referee detection, and penalty box detection. Generation of summaries and initial training required for adaptation of parameters are explained in Section IV. Experimental results over more than 13 hours of soccer video from different regions of the world and the temporal performance of the system are discussed in Section V.

II. Low-Level Analysis for Cinematic Feature Extraction

This section explains the algorithms for low-level cinematic feature extraction, such as shot boundary detection, shot classification, and slow-motion replay detection. Since both shot boundary detector and shot classifier rely on accurate detection of soccer field region in each frame, we start by presenting our robust dominant color region detection algorithm.

A. Robust Dominant Color Region Detection

A soccer field has one distinct dominant color (a tone of green) that may vary from stadium to stadium, and also due to weather and lighting conditions within the same stadium as shown in Fig. 2. Therefore, we do not assume any specific value

for the color of the field in our framework. Our only assumption is the existence of *a single dominant color* that indicates the soccer field. The statistics of this dominant color, in the HSI (hue-saturation-intensity) space, are learned by the system at start-up, and then automatically updated to adapt to temporal variations.

The dominant field color is described by the mean value of each color component, which are computed around their respective histogram peaks. The computation involves determination of the peak index, i_{peak}, for each histogram, which may be obtained from one or more frames. Then, an interval, $[i_{\min}, i_{\max}]$, about each peak is defined, where $i_{\min}$ and $i_{\max}$ refer to the minimum and maximum indices of the interval, respectively, that satisfy the conditions in Eqs. (1)–(6), where H refers to color histogram. The conditions define the minimum (maximum) index as the smallest (largest) index to the left (right) of, including, the peak that has a predefined number of pixels. In our implementation, we fixed this minimum number as 20% of the peak count, i.e., $K = 0.2$. Finally, the mean color in the detected interval is computed for each color component by (7)

$$H[i_{\min}] \geq K * H[i_{peak}] \tag{1}$$

$$H[i_{\min} - 1] < K * H[i_{peak}] \tag{2}$$

$$H[i_{\max}] \geq K * H[i_{peak}] \tag{3}$$

$$H[i_{\max} + 1] < K * H[i_{peak}] \tag{4}$$

$$i_{\min} \leq i_{peak} \tag{5}$$

$$i_{\max} \geq i_{peak} \tag{6}$$

$$\text{Color mean } = \frac{\sum_{i=i_{\min}}^{i_{\max}} H[i] * i}{\sum_{i=i_{\min}}^{i_{\max}} H[i]} * Q_{size}. \tag{7}$$

In (7), Q_{size} is the quantization size, and is used to convert an index to a color value. It assumes different values for hue, saturation, and intensity.

Field colored pixels in each frame are detected by finding the distance of each pixel to the mean color by the *robust* cylindrical metric [19]. Since the algorithm works in the HSI space, achromaticity must be handled with care. If the estimated saturation and intensity means fall in the achromatic region, only intensity distance in Eq. (8) is computed for *achromatic* pixels. Otherwise, both (8) and (9) are employed for *chromatic* pixels in each frame

$$d_{intensity}(j) = |I_j - \overline{I}| \tag{8}$$

$$d_{chroma}(j) = \sqrt{(S_j)^2 + (\overline{S})^2 - 2S_j\overline{S}\cos(\theta(j))} \tag{9}$$

$$d_{cylindrical}(j) = \sqrt{(d_{intensity}(j))^2 + (d_{chroma}(j))^2} \tag{10}$$

$$\theta(j) = \begin{cases} \Delta(j) & \text{if } \Delta(j) <= 180^\circ \\ 360^\circ - \Delta(j) & \text{otherwise} \end{cases} \tag{11}$$

$$\Delta(j) = |\overline{Hue} - Hue_j|. \tag{12}$$

In the equations, Hue, S, and I refer to hue, saturation and intensity, respectively, j is the jth pixel, $\overline{A}$ indicates the dominant color value for the color component A, and θ is defined in (11). The field region is defined as those pixels having

$$d_{cylindrical} < T_{color} \tag{13}$$

where T_{color} is a pre-defined threshold value, and its optimum value for a particular video can be adjusted. T_{color} value that is set after observing only a few seconds of a video provides a robust segmentation in the entire clip, which is usually more than 45 min, thanks to our automatic update of the color statistics, which is explained in Section IV-B.

B. Shot Boundary Detection

Shot boundary detection is usually the first step in generic video processing. Although it has a long research history, it is not a completely solved problem [20]. Sports video is arguably one of the most challenging domains for robust shot boundary detection due to following observations: 1) There is strong color correlation between sports video shots that usually does not occur in a generic video. The reason for this is the possible existence of a single dominant color background, such as the soccer field, in successive shots. Hence, a shot change may not result in a significant difference in the frame histograms. 2) Sports video is characterized by large camera and object motions. Pans and zooms are extensively used to track and focus moving game objects, respectively. Thus, existing shot boundary detectors that rely on change detection statistics are not suitable for sports video. 3) A sports video clip almost always contains both cuts and gradual transitions, such as wipes and dissolves. Therefore, reliable detection of all types of shot boundaries is essential. In addition, we also would like to have a *real-time* performance that requires the use of local rather than global video statistics and robustness to spatial downsampling for speed purposes.

In the proposed algorithm, we take the first observation into account by introducing a new feature, *the absolute difference between two frames in their ratios of dominant (grass) colored pixels to total number of pixels* denoted by G_d. Computation of G_d between the ith and $(i-k)$th frames is given by (14), where G_i represents the grass colored pixel ratio in the ith frame. As the second feature, we use *the difference in color histogram similarity*, H_d, which is computed by (15). The similarity between two histograms is measured by histogram intersection in (16), where the similarity between the ith and the $(i-k)$th frames, $HI(i, k)$, is computed. In the same equation, N denotes the number of color components, and is three in our case, B_m is the number of bins in the histogram of the mth color component, and H_i^m is the *normalized* histogram of the ith frame for the same color component. The algorithm uses different k values in (14)–(16) to detect cut-type boundaries and gradual transitions. Since cuts are instant transitions, $k = 1$ detects cuts, while we check a range of k values, $k > 1$, instead of a single k to locate gradual transitions (we have determined the upper bound for k to be 5)

$$G_d(i, k) = |G_i - G_{i-k}| \tag{14}$$

$$H_d(i, k) = |HI(i, k) - HI(i-k, k)| \tag{15}$$

$$HI(i, k) = \frac{1}{N}\sum_{m=1}^{N}\sum_{j=0}^{B_m-1} \min\left(H_i^m[j], H_{i-k}^m[j]\right). \tag{16}$$

A shot boundary is determined by comparing H_d and G_d with a set of thresholds. A novel feature of the proposed method, in addition to **the introduction of G_d as a new feature**, is **the adaptive change of the thresholds on H_d**. When a sports video shot corresponds to out of field or close-up views (the definitions of both will be given in Section II-C), the number of field colored pixels will be very low and shot properties will be similar to a generic video shot. In such cases, the problem is the same as generic shot boundary detection; hence, we use only H_d with a high threshold. In the situations where the field is visible, we use both H_d and G_d, but using a lower threshold for H_d. Thus, we define four thresholds for shot boundary detection: T_H^{Low}, T_H^{High}, T_G, and $T_{lowGrass}$. The first two thresholds are the low and high thresholds for H_d, and T_G is the threshold for G_d. The last parameter, $T_{lowGrass}$, is essentially a rough estimate for low grass ratio and determines when the conditions change from field view to out of field or close-up view. That is, when grass colored pixel ratio in the ith frame, G_i, is lower than $T_{lowGrass}$, the algorithm compares H_d against T_H^{High}; otherwise, T_H^{Low} is picked up for the comparison. The optimum values for these thresholds can be set for each sport type after a learning stage. Once the thresholds are set, the algorithm needs only to compute local statistics, and runs in *real-time*. Furthermore, the proposed algorithm is robust to spatial downsampling since both G_d and H_d are size-invariant. In Section V, we will present our results on 4×4 spatially downsampled video.

C. Shot Classification

Shot class information, when combined with other features, conveys interesting semantic cues. Motivated by this observation, we classify soccer shots into three classes [21]: 1) Long shots, 2) In-field medium shots, and 3) Out of field or close-up shots. The definitions and characteristics of each class are given below:

- **Long Shot:** A long shot displays the global view of the field as shown in Fig. 3(a) and (b); hence, a long shot serves for accurate localization of the events on the field.
- **In-Field Medium Shot:** A medium shot, where a whole human body is usually visible, is a zoomed-in view of a specific part of the field as in Fig 3(c) and (d). Although the occurrence of a single isolated medium shot between long shots corresponds to a play, a group of nearby medium shots usually indicates a break in the game. Furthermore, a replay is more likely to be shown as a medium shot than as either of the other two shot types.
- **Close-Up Shot:** A close-up shot shows the above-waist view of one person [Fig. 3(e)]. In general, the occurrence of a close-up shot indicates a break in the game.
- **Out of Field Shot:** The audience, coach, and other shots are denoted as out of field shots [Fig. 3(f)]. Similar to close-ups, an out of field shot often indicates a break in the game. We analyze both out of field and close-up shots in the same category due to their similar semantic meaning.

Classification of a shot into one of the above three classes is based on spatial features. Therefore, shot class can be determined from a single key frame or from a set of frames selected according to a certain criteria. Due to the computational simplicity of our algorithm, we find the class of every frame in a shot and assign the shot class to the label of the majority of frames. In order to find the frame view, frame grass colored pixel ratio, G, is computed. In [9], an intuitive approach is used, where a low G value in a frame corresponds to a close-up or out of field view, while high G value indicates that the frame is a long view, and in between, a medium view is selected. Although the accuracy of the above simple algorithm is sufficient for some applications, such as play-break detection in [9], it has been proven to be insufficient for our application, which uses these low-level results to reach higher level semantics. By using only grass colored pixel ratio, medium shots with high G value will be mislabeled as long shots. The error rate due to this approach depends on the broadcasting style and it usually reaches intolerable levels for the employment of higher level algorithms in Section III.

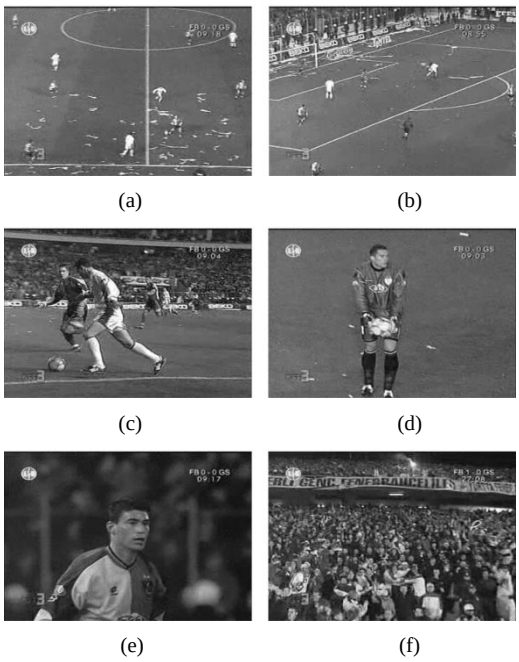


Fig. 3. View types in soccer: (a), (b) Long view, (c), (d) in-field medium view, (e) close-up view, and (f) out of field view.

We propose a compute-easy, yet very efficient, cinematographic algorithm for the frames with a high G value. We define regions by using *Golden Section* spatial composition rule [22], [23], which suggests dividing up the screen in $3:5:3$ proportion in both directions, and positioning the main subjects on the intersection points of these lines. We have revised this rule for soccer video, and divide the *grass region box* instead of the whole frame. *Grass region box* can be defined as the minimum bounding rectangle (MBR), or a scaled version of it, of grass colored pixels. In Fig. 4, the examples of the regions obtained by *Golden Section* rule is displayed on several medium and long views. In the regions R_1, R_2, and R_3 in Fig. 4(d) and (f), we have defined 8 features to measure the distribution of the grass colored pixels in medium and long views, and found the two features below the most distinguishing:

- G_{R_2}: The grass colored pixel ratio in the second region
- R_{diff}: The mean value of the absolute grass color pixel differences between R_1 and R_2, and between R_2 and R_3

$$R_{diff} = \tfrac{1}{2}\left\{|G_{R_1} - G_{R_2}| + |G_{R_2} - G_{R_3}|\right\}. \qquad (17)$$

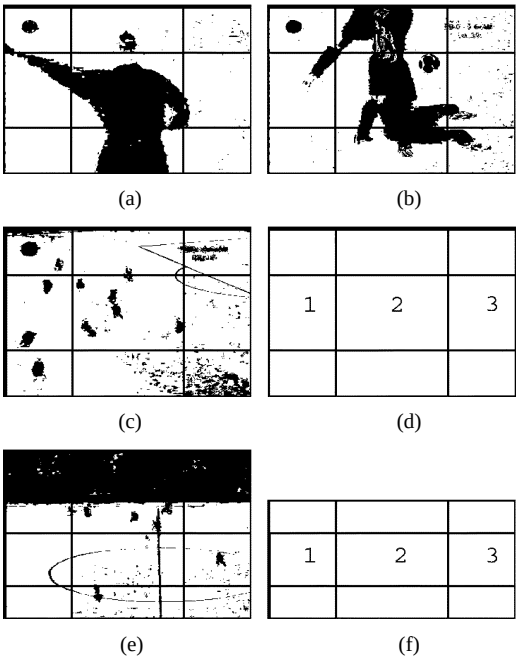


Fig. 4. Examples of Golden Section spatial composition in (a), (b) medium and (c)–(e) long views, the resulting grass region boxes and the regions are shown in (d) and (f) for (a)–(c) and (e), respectively.

We employ a Bayesian classifier using the above two features. A Bayesian classifier assigns the feature vector x, which is assumed to have a Gaussian distribution, to the class that maximizes the discriminant function $g(x)$ [24]:

$$g_i(x) = -\tfrac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) + \ln P(w_i) + c_i \quad (18)$$

$$c_i = -\ln(2\pi) - (1/2)\ln|\Sigma_i| \quad (19)$$

$$x = |G_{R_2} \quad R_{diff}|^T. \quad (20)$$

The mean, μ, and the covariance matrices, Σ, for long and medium views have been computed from a total of 50 frames (equal number of frames from each class) selected as the training set, and the class probability values are assumed to be equal.

The flowchart of the proposed shot classification algorithm is shown in Fig. 5. The first stage uses G value and two thresholds, $T_{closeUp}$ and T_{medium} to determine the frame view label. These two thresholds are roughly initialized to 0.1 and 0.4 at the start of the system, and as the system collects more data, they are updated to the minimum of the grass colored pixel ratio, G, histogram as suggested in [9]. When $G > T_{medium}$, the algorithm determines the frame view by using our novel cinematographic features in (18)–(20).

D. Slow-Motion Replay Detection

Replays in sports broadcasts are excellent locators of semantically important segments for high-level video processing. Several slow-motion replay detectors for compressed and spatial domains exist in the literature [25]–[27].[2] Since we only need to determine if a given shot consists of a slow-motion segment, the zero crossing measure proposed in [26] has proved to be sufficient for our application.[3] Zero crossing measure evaluates the amplitude of the fluctuations in the frame differences ($D(t)$ values) within a window of length L. The frame difference for the frame at the discrete time t, is denoted as $D(t)$ and computed by (21), where M and N are the width and the height of the frames, respectively. The $D(t)$ values of a sample slow-motion shot are shown in Fig. 6 to exemplify the large fluctuations. To compensate for the shot motion contribution to $D(t)$, the mean of $D(t)$, $\overline{D}(t)$, in the processed window is subtracted from each $D(t)$ value in the same window. The amplitude of the fluctuations affect the zero crossing value through the quantization levels, θ_k. The window length, L, is compensated by β, which defines the threshold for the number of fluctuations in the processed window. Finally, the number of zero crossings, $p_{zc}(t)$, is defined through (21)–(24) as the largest index value, k, for which $Z_c(t, \theta_k) > \beta$

$$D(t) = \frac{1}{M*N}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(I_t(i,j) - I_{t-1}(i,j)\right)^2 \quad (21)$$

$$Z_c(t,\theta_k) = \sum_{i=1}^{L-1} f(D(t-i) - \overline{D}(t), \quad D(t-i-1) - \overline{D}(t), \theta_k) \quad (22)$$

$$p_{zc}(t) = k \qquad \text{if } Z_c(t,\theta_k) > \beta \quad (23)$$

$$f(x,y,\theta) = \begin{cases} 1 & x \geq \theta \text{ and } y \leq -\theta \\ & \text{or } x \leq -\theta \text{ and } y \geq \theta \\ 0 & \text{otherwise.} \end{cases} \quad (24)$$

In [26], the details about the quantization levels, θ_k, the value of the threshold on zero crossing value, β, and the window length, L, are not given. In order to accurately determine those thresholds, we make several observations: First, the subtraction of $D(t)$ values by their average in a window, $\overline{D}(t)$, compensates the motion effect only to some extent. For example, player motion in a close-up shot and the camera motion in a long shot may cause large fluctuations in $D(t)$. Therefore, we adaptively change the quantization step size by the average motion content, which is assumed to be proportional to the mean value of $D(t)$ in the entire shot. The quantization step size ranges from 1 to 4, i.e., a shot with low motion content will be compared against $\theta_0 = 1$, $\theta_1 = 2$, and so on, while a shot with a very high motion content will have $\theta_0 = 4$, $\theta_1 = 8$, and so on. The largest index, k, for slow-motion decision is empirically found to be 100. In addition to frame repeats, we also observe that frame interpolation and high-speed cameras may be employed to generate slow-motions. Therefore, as shown in Fig. 6, $D(t)$ values may result in minima instead of zeros during slow-motions. There is a relation between the number of repeated (or interpolated) frames, denoted as f, and the parameters β (the zero crossing threshold in a window) and L (window length)

[2]Reference [27] improves [26] by detecting logo transitions. Since the use of logo transitions before and after replays is broadcaster-dependent, [26] is a more generic algorithm.

[3]In [26], several other features are also proposed to both locate the slow-motions and to find other fields, such as still frames and normal motion replays, in a sports video.

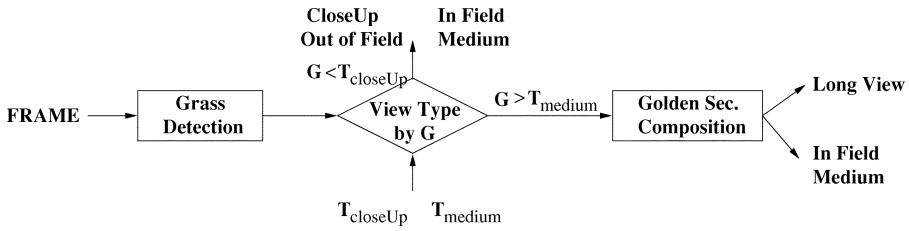
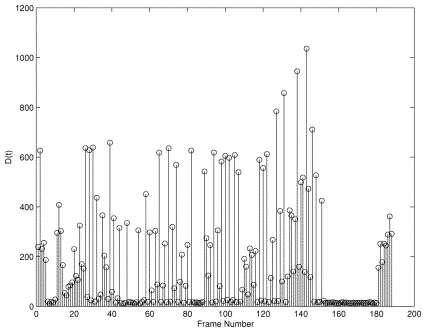
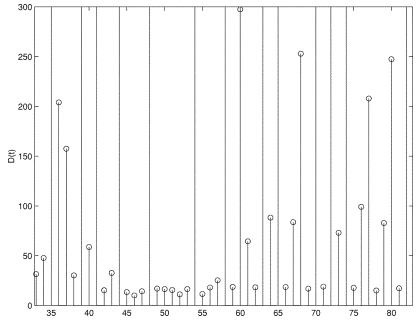


Fig. 5. Flowchart of the shot type (view) classification algorithm.



(a)



(b)

Fig. 6. (a) Fluctuations of $D(t)$ values in a slow-motion replay shot and (b) the zoomed-in view for the indices between 33 and 82.

that can be expressed as $L > f + \beta$. However, f value is not usually constant as exemplified in Fig. 6(b), where it is equal to 1 or 2 before index 45, 3 starting at index 45, and 5 around index 50. Furthermore, a large value for L is not usually favorable, since the selected window should not include too many frames having normal motion. Therefore, we have selected 7 for L, and 1 for β.

III. Soccer Event and Object Detection

Detection of certain events and objects in a soccer game enables generation of more concise and semantically rich summaries. Since goals are arguably the most significant event in soccer, we propose a novel goal detection algorithm in Section III-A. The proposed goal detector employs *only cinematic features* and runs in *real-time*. Goals, however, are not the only interesting events in a soccer game. Controversial calls, such as red-yellow cards and penalties (medium and close-up shots involving referees), and plays inside the penalty box, such as shots and saves, are also important for summarization and browsing. Therefore, we also develop novel algorithms for referee and penalty box detection that are presented in Sections III-B and III-C, respectively. We use referee and penalty box detection results to generate summaries as a function of these descriptors.

A. Goal Detection

A goal is scored when the whole of the ball passes over the goal line, between the goal posts and under the crossbar [28]. Unfortunately, it is difficult to verify these conditions automatically and reliably by video processing algorithms. However, occurrence of a goal is generally followed by a special pattern of cinematic features, which is what we exploit in our proposed goal detection algorithm. A goal event leads to a break in the game. During this break, the producers convey the emotions on the field to the TV audience and show one or more replay(s) for a better visual experience. The emotions are captured by one or more close-up views of the actors of the goal event, such as the scorer and the goalie, and by shots of the audience celebrating the goal. For a better visual experience, several slow-motion replays of the goal event from different camera positions are shown. Then, the restart of the game is usually captured by a long shot. Between the long shot resulting in the goal event and the long shot that shows the restart of the game, we define *a cinematic template* that should satisfy the following requirements.

- *Duration of the break:* A break due to a goal lasts no less than 30 and no more than 120 seconds.
- *The occurrence of* **at least one** *close-up/out of field shot*: This shot may either be a close-up of a player or out of field view of the audience.
- *The existence of* **at least one** *slow-motion replay shot*: The goal play is always replayed one or more times.
- *The relative position of the replay shot:* The replay shot(s) follow the close-up/out of field shot(s).

In Fig. 7, the instantiation of the template is demonstrated for the first goal in *Spain1* sequence of MPEG-7 data set, where the break lasts for 54 sec. In order to detect goals, for every slow-motion replay shot, the system finds the long shots that define the start and the end of the corresponding break. These long shots must indicate a play that is determined by a simple duration constraint, i.e., long shots of short duration are labeled as breaks. Finally, the conditions of the template are verified to detect goals. The proposed "cinematic template" models goal events very well, and the detection runs in real-time with a very high recall rate. Other interesting events may also fit this template although not as consistently as goals. The addition of such segments in the summaries may even be desirable since each such segment consists of interesting segments. Therefore, the recall rate for this algorithm is much more important than the precision rate, since the users will not be tolerant to missing goals, but may enjoy watching interesting nongoal events.

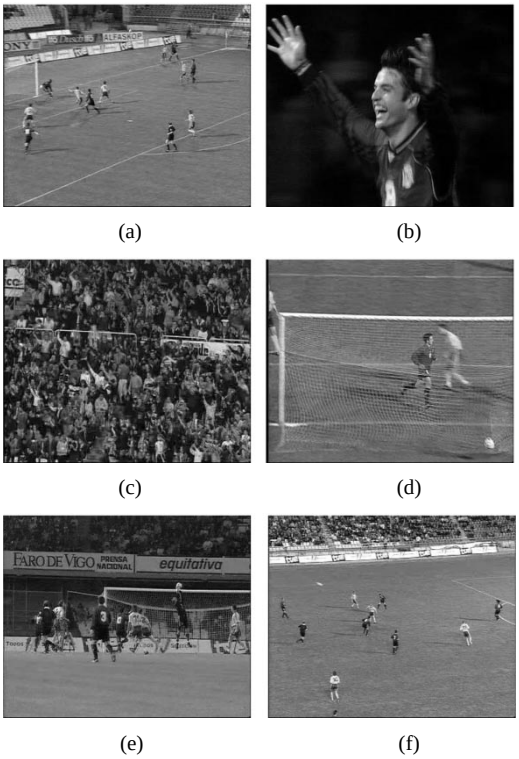


Fig. 7. The broadcast of the first goal in *Spain1*: (a) long view of the actual goal play, (b) player close-up, (c) audience, (d) the first replay, (e) the third replay, and (f) long view of the start of the new play.

B. Referee Detection

Referees in soccer games wear distinguishable colored uniforms from those of the two teams on the field. Therefore, a variation of our dominant color region detection algorithm in Section II-A can be used to detect referee regions. We assume that there is, if any, a single referee *in a medium or out of field/close-up shot* (we do not search for a referee in a long shot). Then, the horizontal and vertical projections [29] of the feature pixels can be used to accurately locate the referee region. The peak of the horizontal and the vertical projections and the spread around the peaks are employed to compute the rectangle parameters surrounding the referee region, hereinafter "MBR_{ref}." MBR_{ref} coordinates are defined to be the first projection coordinates at both sides of the peak index without enough pixels, which is assumed to be 20% of the peak projection. In Fig. 8, an example frame, the referee pixels in that frame, the horizontal and vertical projections of the referee region, and the resulting referee MBR_{ref} are shown.

The decision about the existence of the referee in the current frame is based on the following size-invariant *shape* descriptors.

- *The ratio of the area of the MBR_{ref} to the frame area*: A low value indicates that the current frame does not contain a referee.
- *MBR_{ref} aspect ratio (width/height)*: Frames with aberrant MBR_{ref} aspect ratio values are discarded. In our system, we consider aspect ratio values outside (0.2, 1.8) interval as outliers.
- *Feature pixel ratio in the MBR_{ref}*: This feature approximates the compactness of MBR_{ref}, higher compactness values, i.e., higher referee pixel ratios, are favored.

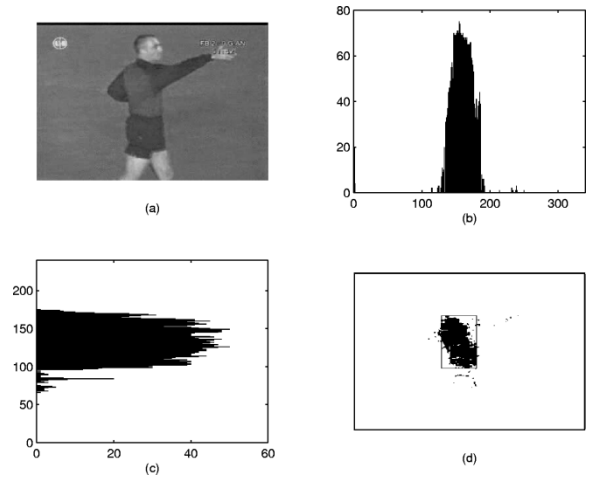


Fig. 8. Referee in the input frame (a) is detected by using the horizontal (b) and the vertical (c) projections of the binary referee mask image (d).

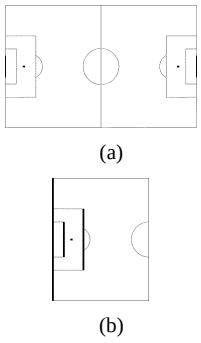


Fig. 9. (a) Soccer field model and (b) three highlighted parallel lines around goal area.

- *The ratio of the number of feature pixels in the MBR_{ref} to that of the outside*: It measures the correctness of the single referee assumption. When this ratio is low, the single referee assumption does not hold, and the frame is discarded.

The proposed approach for referee detection runs very fast, and it is robust to spatial downsampling. We have obtained comparable results for original (352 $\times$ 240 or 352 $\times$ 288), and for 2 $\times$ 2 and 4 $\times$ 4 spatially downsampled frames.

C. Penalty Box Detection

Field lines *in a long view* can be used to localize the view and/or register the current frame on the standard field model. In this section, we reduce the penalty box detection problem to the search for three parallel lines. In Fig. 9(a), a view of the whole soccer field is shown, and **three parallel field lines**, shown in bold in Fig. 9(b), become visible when the action occurs around or within one of the penalty boxes. This observation yields a robust method for penalty box detection, and it is arguably more accurate than the goal post detection proposed in [3] for a similar analysis, since goal post views are likely to include cluttered background pixels that cause problems for Hough transform.

To detect three lines, we use the grass detection result in Section II-A. To limit the operating region to the field pixels, we compute a mask image from the grass colored pixels, displayed in Fig. 10(b). The mask is obtained by first computing a scaled version of the grass MBR, drawn on the same figure, and then,

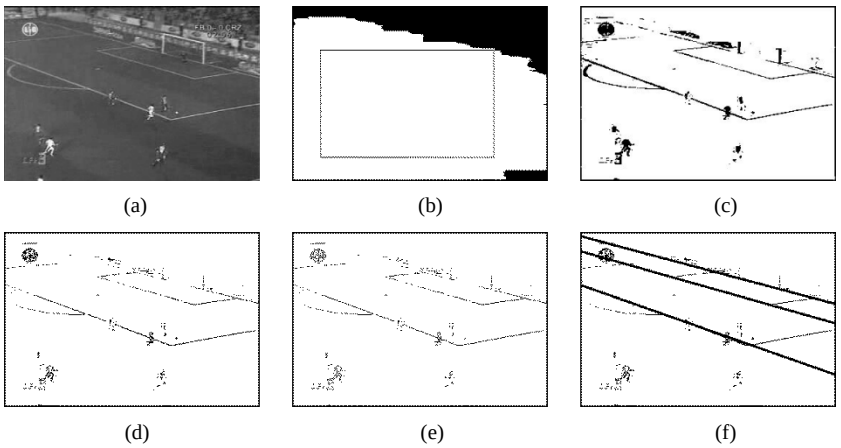


Fig. 10. Penalty box detection, (a) the input frame, (b) the field mask, (c) grass/nongrass image in the field region, (d) the pixels in (c) with high gradient, (e) image after thinning, and (f) three detected lines.

by including all field regions that have enough pixels inside the computed rectangle. As shown in Fig. 10(c), nongrass pixels may be due to lines and players on the field. To detect line pixels, we use edge response, defined as the pixel response to the 3×3 Laplacian mask in (25). The pixels with the highest edge response, the threshold of which is automatically determined from the histogram of the gradient magnitudes, are defined as line pixels. The resulting line pixels after the Laplacian mask operation and the image after thinning are shown in Fig. 10(d) and (e), respectively

$$h = \begin{vmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{vmatrix}. \tag{25}$$

Then, three parallel lines are detected by Hough transform that employs *size*, *distance* and *parallelism* constraints. As shown in Fig. 9(b), the line in the middle is *the shortest line*, and it has *a shorter distance to the goal line (outer line)* than to the penalty line (inner line). The detected three lines of the penalty box in Fig. 10(a) are shown in Fig. 10(f).

IV. Summarization and Adaptation of Parameters

In this section, we explain the generation and presentation of summary clips, and the training details for the algorithms. The proposed framework provides three types of summaries as all slow-motion segments, all goal events, and the extension of the two with object-based features. As explained below, these summaries can be customized by user preferences. Training of the system for a particular game can be performed in a very short time during the pre-game broadcasts.

A. Summarization and Presentation

The proposed framework includes three types of summaries: 1) All slow-motion replay shots in a game, 2) all goals in the same game, and 3) the extension of the two with object-based features. The first two types of summaries are based solely on cinematic features, and are generated in real-time, while the last type uses referee and penalty box detection results.

Slow-motion summaries are generated by shot boundary, shot class, and slow-motion replay features, and consist of slow-motion shots. Depending on the requirements, they may also consist of all shots in a pre-defined time window around each replay, or, instead, they can include only the closest long shot before each replay in the summary, since the closest long shot is likely to include the corresponding action in normal motion. As explained in Section III-A, goals are detected in a cinematic template. Therefore, *goal summaries* consist of the shots in the detected template, or in its customized version, for each goal. Finally, summaries with referee and penalty box objects are generated. To determine if a slow-motion shot involve referee and/or a penalty box, we select segments of interest for each. The segments of interest include close nonlong shots around the corresponding replay for referee detection, and one or more closest long shots *before* the replay for penalty box detection. Then, object-based summaries include those shots with the detected object, in addition to the replays.

Interaction with the system is an important feature for the customization of the generated summaries by the user preferences and the requirements. The users may want to stream only certain parts of the summaries due to the available bandwidth, such as in a wireless environment, or their time requirements. The proposed framework enables such interactivity due to the rich detail in the generated summaries. For example, the system can offer the user a menu where they can choose to skip the shots for player-audience views, or even replays of the goals against their favorite team. Similarly, the users can customize the system to skip/display all shots where the referee is visible, to watch only goal area positions, or to stream the scenes where both the goal area and the referee appear.

B. Adaptation of Parameters

In this section, we explain the system parameter adaptation required for each game. This type of training is different than domain-based training, which refers to one-time training for soccer domain. The algorithms for shot boundary detection, slow-motion replay detection, and penalty box detection use thresholds that apply to any soccer game. On the other hand, T_{color} in dominant color region detection, $T_{closeUp}$ and T_{medium} in shot classification, and referee color statistics vary in each game and their fine-tuning to a particular video is the focus of this section. The training stage can be performed in a very short time, usually

TABLE I
THE NAMES AND THE LENGTH OF THE CLIPS IN THE DATABASE (1 REFERS TO THE FIRST HALF)

Names and the length (min:sec) of the clips
Korea1(54:53), Gant1(48:22), Gant2(46:59), Spain1(14:58), Mlt1(49:57), Mlt2(49:09), Ts2(48:09), Den1(47:27), Den2(47:51), Rize1(47:36), Rize2(49:15), DBak1(46:25), DBak2(48:25), Goz1(47:30), Goz2(48:56), Ant1(45:43), Gs1(49:05)

during pre-game broadcasts to make the system ready before the game starts.

T_{color} value in (13) is interactively set after observing only a few seconds of a video. A similar threshold used for referee detection is also needed to be adjusted. The location of the referee can be specified through a user-friendly interface. Even the use of only short video segments in both situations provides a robust segmentation in the entire clip thanks to the algorithm's automatic update of the color statistics that compensates time-varying nature of the field color statistics, especially due to changing weather and/or shading. The adaptation to the temporal variations is achieved by collecting color statistics of each pixel that satisfies Eq. (13), where $d_{cylindrical}$ is compared with a larger threshold value. That means, in addition to the field pixels, the close nonfield pixels are included to the field histogram computation. When the system needs an update, the collected statistics are used to estimate the new mean color values using Eq. (7).

The thresholds $T_{closeUp}$ and T_{medium} are initialized to 0.1 and 0.4 at the start of the system, and as the system collects more data, they are updated to the minimum of the grass colored pixel ratio, G, histogram as suggested in [9]. This process is nonsupervised and we have observed that once the exact values of $T_{closeUp}$ and T_{medium} are learned in a video clip belonging to a particular broadcaster, the same values hold for a different video of the same broadcaster.

V. RESULTS

We have rigorously tested the proposed algorithms on a data set of more than 13 hours (800 min.) of soccer video. The database is composed of 17 MPEG-1 clips, 16 of which in 352×240 resolution at 30 fps,[4] and one (*Spain1* sequence from MPEG-7 set) in 352×288 resolution at 25 fps. Table I shows the name and the length of each clip in the database. We have used several short clips from *Ant1* and *Gs1* sequences for training. The segments used for training are omitted from the test set; hence, neither sequence is used by the goal detector.

A. Results for Low-Level Algorithms

We define two ground truth sets, one for shot boundary detector and shot classifier, and one for slow-motion replay detector. The first set is obtained from *Gant1*, *Korea1*, and *Spain1* sequences, and it consists of 49 minutes of video as shown in Table II. The sequences are not chosen arbitrarily; on the contrary, we intentionally selected the sequences from different countries to demonstrate the robustness of the proposed algorithms to varying cinematic styles.

TABLE II
SHOT BOUNDARY DETECTION RESULTS (G.Tr. = GRADUAL TRANSITIONS)

Sequence	Gant1		Korea1		Spain1		All	
Length	20:53		15:53		12:43		49:29	
Type	Cuts	G.Tr.	Cuts	G.Tr.	Cuts	G.Tr.	Cuts	G.Tr.
Correct	160	29	113	16	47	13	320	58
False	10	5	13	3	6	1	29	9
Miss	0	5	7	2	2	3	9	10
Recall	100	85.3	94.2	88.9	95.9	81.3	97.3	85.3
Precision	94.1	85.3	89.7	84.2	88.7	92.9	91.7	86.6

Each frame in the first set is downsampled, without low-pass filtering, by a rate of four in both directions to satisfy the real-time constraints, that is, 88×60 (88×72 for *Spain1* sequence) is the actual frame resolution for shot boundary detector and shot classifier. In Table II, the recall and the precision rates of the shot boundary detector are given for each sequence and for the whole set. The performance of the algorithm for cut-type boundaries and gradual transitions is tabulated separately. Overall, the algorithm achieves 97.3% recall and 91.7% precision rates for cut-type boundaries. On the same set at full resolution, a generic cut-detector [30], which comfortably generates high recall and precision rates (greater than 95%) for nonsports video, has resulted in 75.6% recall and 96.8% precision rates. A generic algorithm, as expected, misses many shot boundaries due to the strong color correlation between sports video shots. The precision rate at the resulting recall value does not have a practical use. The proposed algorithm also reliably detects gradual transitions, which refer to *wipes* for *Gant1*, *wipes* and *dissolves* for *Spain1*, and *other editing effects* for *Korea1*. On the average, the algorithm works at 85.3% recall and 86.6% precision rates. The highest recall rate for gradual transitions is achieved for *Korea1* sequence, where the editing effects cause more fluctuations in the features than wipes do. Gradual transitions are difficult, if not impossible, to detect when they occur between two long shots or between a long and a medium shot with a high grass ratio.

The accuracy of the shot classification algorithm, which uses the same 88×60 or 88×72 frames as shot boundary detector, is shown in Table III.[5] For each sequence, we provide two results, one by using only grass colored pixel ratio, G, and the other by using both G and the proposed features, G_{R_2} and R_{diff}. Our

[4] 29.97 to be exact.

[5] Only correctly detected shots in Table II are used for view classification, and four shots in that set that deviate from each shot type due to the missing or false boundaries are discarded.

TABLE III
VIEW CLASSIFICATION RESULTS FOR THREE TEST SEQUENCES, (METHOD G USES ONLY GRASS MEASURE, WHILE METHOD P IS THE PROPOSED METHOD)

Sequence	Gant1		Korea1		Spain1		All	
Method	G	P	G	P	G	P	G	P
# of Shots	188	188	128	128	58	58	374	374
Correct	131	164	106	114	47	55	284	333
False	57	24	22	14	11	3	90	41
Accuracy(%)	69.7	87.2	82.8	89.1	81.0	94.8	75.9	89.0

TABLE IV
SLOW-MOTION REPLAY DETECTION RESULTS, OVERALL 85.2% PRECISION, AND 80% RECALL RATES

Sequence Name	Length in (min:sec)	Length in Frames	# of Replay Shots	Correct	False	Miss
Gant1	(21:28)	38,632	18	15	3	3
Spain1	(12:43)	19,084	7	5	0	2
Ts2	(15:06)	27,186	16	14	1	2
Ant1	(26:50)	48,300	13	10	4	3
Korea1	(17:25)	31,340	11	8	1	3
Total	(93:32)	164,542	65	52	9	13

results for *Korea1* and *Spain1* by only G are very close to the reported results on the same set in [9]. By introducing two new features, G_{R_2} and R_{diff}, we are able to obtain 17.5%, 6.3%, and 13.8% improvement in *Gant1*, *Korea1*, and *Spain1* sequences, respectively. The results clearly indicate the effectiveness and the robustness of the proposed algorithm for different cinematographic styles.

The ground truth for slow-motion replays includes two new sequences, *Ant1* and *Ts2*, making the length of the set 93 min, which is approximately equal to a complete soccer game, as shown in Table IV. The slow-motion detector uses frames at *full resolution*, and has detected 52 of 65 replay shots, 80.0% recall rate, and incorrectly labeled 9 normal motion shots, 85.2% precision rate, as replays. These results are somewhat worse than the reported results, 100% recall without explicit precision rate, in [26]. In addition to using only $D(t)$, resolution and compressed format can also be counted for the difference since the detector is sensitive to resolution and precise pixel values. The content features, such as abrupt and fast camera motions in long shots and irregular object motion in close-ups, are the main reasons for false positives (In [26], only one soccer game that is less than a minute is used). Overall, the recall-precision rates in slow-motion detection are quite satisfactory.

B. Results for High-Level Analysis and Summarization

Goals are detected in 15 test sequences in the database. Each sequence, in full length, is processed to locate shot boundaries, shot types, and replays. When a replay is found, goal detector computes the cinematic template features to find goals. The performance of the goal detector is demonstrated in Table V for each sequence and for the whole set. The proposed algorithm runs *in real-time*, and, on the average, achieves a 90.0% recall and 45.8% precision rates, which are quite satisfactory

TABLE V
THE DISTRIBUTION OF GOAL DETECTION RESULTS FOR EACH SEQUENCE

Sequence	Correct	False	Miss	Sequence	Correct	False	Miss
Gant1	2	2	0	Spain1	2	0	0
Gant2	3	3	0	Korea1	2	2	1
Mlt1	2	3	0	Den1	2	2	0
Mlt2	1	1	1	Den2	1	3	0
DBak1	1	1	0	Goz1	1	2	0
DBak2	2	1	1	Goz2	2	2	0
Rize1	2	6	0	Ts2	3	0	0
Rize2	1	4	0	**TOTAL**	**27**	**32**	**3**

TABLE VI
THE STATISTICS ABOUT THE APPEARANCE OF REFEREE FOR SOME SEMANTIC EVENTS

	Yellow/Red Cards	Penalties	Free-Kicks
Total	19	3	8
Referee Appears	19	3	5
Detected	16	3	5
Recall(%)	84.2	100	100
Confidence(%)	100	100	62.5

for a real-time system. As explained in Section III-A, the recall rate for this algorithm is much more important than the precision rate, since the user can always fast-forward nongoal events or may even enjoy watching interesting nongoal events ("interesting" due to the use of replays) in the summary. Two of the misses in Table V are due to the inaccuracies in the extracted shot-based features, and the miss in *Mlt1*, where the replay shot is broadcast minutes after the goal, is due to the deviation from the goal model. The number of nongoal events in the goal summaries is proportional to the frequency of the breaks in the game. The frequent breaks due to fouls, offsides, shots to goal, etc. with one or more slow-motion shots may generate cinematic templates similar to that of a goal. The inaccuracies in shot boundaries, shot types, and replay labels may also contribute to the same situation.

In Section III, we explained that the existence of *referee* and *penalty box* in a summary segment, which, by definition, also contains a slow-motion shot, may correspond to certain events. Then, the user can browse summaries by these *object-based features*. The recall rate of and the confidence with referee and penalty box detection are specified for a set of semantic events in Tables VI and VII, where *recall* rate measures the accuracy of the proposed algorithms, and the *confidence* value is defined as the ratio of the number of events with that object to the total number of such events in the database, and it indicates the applicability of the corresponding object-based feature to browsing a certain event. For example, the confidence of observing a referee in a free kick event is 62.5%, meaning that the referee feature may not be useful for browsing free kicks. On the other hand, the existence of both objects is necessary for a penalty event due

TABLE VII
THE STATISTICS ABOUT THE APPEARANCE OF PENALTY BOX (PBOX) FOR SOME SEMANTIC EVENTS

	Shots/Saves	Penalties	Free-Kicks
Total	50	3	8
Penalty Box Appears	49	3	8
Detected	41	3	8
Recall(%)	83.7	100	100
Confidence(%)	98.0	100	100

TABLE VIII
THE TIME COST OF THE LOW-LEVEL ALGORITHMS

Algorithm	Downsamp. Rate	Time (ms)
Grass and Mask Detection	4x4	9.02
Shot Bound. and View Classification	4x4	4.47
Slow-Motion	None	23.4

to their high confidence values. In Tables VI and VII, the first row shows the total number of a specific event in the summaries. Then, the second row shows the number of events where the referee and/or three penalty box lines are visible. In the third row, the number of detected events is given. Recall rates in the second columns of both Tables VI and VII are lower than those of the other events. For the former, the misses are due to referee's occlusion by other players, and for the latter, abrupt camera movement during a high activity prevents reliable penalty box detection. Finally, it should be noted that the proposed features and their statistics are used for browsing purposes, not for detecting such nongoal events; hence, precision rates are not meaningful.

The compression rate for the summaries varies with the requested format. On the average, 12.78% of a game is included to the summaries of all slow-motion segments, while the summaries consisting of all goals, including all detected nongoal events, only account for 4.68%, of a complete soccer game. These rates correspond to the summaries that are less than 12 and 5 min, respectively, of an approximately 90-min game.

C. Temporal Performance

The processing time per frame for each low-level algorithm at the specified downsampling rate is given in Table VIII. RGB to HSI color transformation required by grass detection limits the maximum frame size; hence, 4×4 spatial downsampling rates for both shot boundary detection and shot classification algorithms are employed to satisfy the real-time constraints. The accuracy of slow-motion detection algorithm is sensitive to frame size; therefore, no sampling is employed for this algorithm, yet the computation of $D(t)$, is completed in real-time. A commercial system can be implemented by multi-threading where shot boundary detection, shot classification, and slow-motion detection should run in parallel. It is also affordable to implement the first two sequentially, as it was done in our system. In addition to spatial sampling, temporal sampling may also be applied for shot classification without significant performance degradation. In this framework, goals are detected with a delay that is equal to the cinematic template length, which may range from 30 to 120 s as explained in Section III-A.

VI. CONCLUSION

In this paper, a new framework for summarization of soccer video has been introduced. The proposed framework allows real-time event detection by cinematic features, and further filtering of slow-motion replay shots by object-based features for semantic labeling. The implications of the proposed system include real-time streaming of live game summaries, summarization and presentation according to user preferences, and efficient semantic browsing through the summaries, each of which makes the system highly desirable.

The topics for future work include 1) integration of aural and textual features to increase the accuracy of event detection and 2) extension of the proposed framework to different sports, such as football, basketball, and baseball, which require different event and object detection modules.

REFERENCES

[1] S.-F. Chang, "The holy grail of content-based media analysis," *IEEE Multimedia*, vol. 9, pp. 6–10, Apr.–June 2002.
[2] Y. Fu, A. Ekin, A. M. Tekalp, and R. Mehrotra, "Temporal segmentation of video objects for hierarchical object-based motion description," *IEEE Trans. Image Processing*, vol. 11, pp. 135–145, Feb. 2002.
[3] D. Yow, B.-L. Yeo, M. Yeung, and B. Liu, "Analysis and presentation of soccer highlights from digital video," in *Proc. Asian Conf. on Comp. Vision (ACCV)*, 1995.
[4] Y. Gong, L. T. Sin, C. H. Chuan, H.-J. Zhang, and M. Sakauchi, "Automatic parsing of soccer programs," in *Proc. IEEE Int. Conf. Mult. Comput. Syst.*, 1995, pp. 167–174.
[5] S. Intille and A. Bobick, "Recognizing planned, multi-person action," *Comput. Vis. Image Understand.*, vol. 81, no. 3, pp. 414–445, Mar. 2001.
[6] V. Tovinkere and R. J. Qian, "Detecting semantic events in soccer games: Toward a complete solution," in *Proc. IEEE Int. Conf. Mult. Expo (ICME)*, Aug. 2001.
[7] G. S. Pingali, Y. Jean, and I. Carlbom, "Real time tracking for enhanced tennis broadcasts," in *Proc. IEEE Comp. Vision Patt. Rec. (CVPR)*, 1998, pp. 260–265.
[8] A. Gueziec, "Tracking pitches for broadcast television," *IEEE Computer*, vol. 35, pp. 38–43, Mar. 2002.
[9] P. Xu, L. Xie, S.-F. Chang, A. Divakaran, A. Vetro, and H. Sun, "Algorithms and system for segmentation and structure analysis in soccer video," in *Proc. IEEE Int. Conf. Mult. Expo (ICME)*, Aug. 2001.
[10] L. Xie, S.-F. Chang, A. Divakaran, and H. Sun, "Structure analysis of soccer video with hidden Markov models," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 2002.
[11] B. Li and M. I. Sezan, "Event detection and summarization in American football broadcast video," *Proc. SPIE*, vol. 4676, pp. 202–213, Jan. 2002.
[12] R. Leonardi and P. Migliorati, "Semantic indexing of multimedia documents," *IEEE Multimedia*, vol. 9, no. 2, pp. 44–51, Apr.–June 2002.
[13] J. Assfalg, M. Bertini, A. Del Bimbo, W. Nunziati, and P. Pala, "Soccer highlights detection and recognition using HMMs," in *Proc. IEEE Int. Conf. on Mult. and Expo (ICME)*, Aug. 2002.
[14] W. Zhou, A. Vellaikal, and C.-C.J. Kuo, "Rule-based video classification system for basketball video indexing," in *ACM Mult. Conf.*, 2000.
[15] D. Zhong and S.-F. Chang, "Structure analysis of sports video using domain models," in *Proc. IEEE Int. Conf. Mult. Expo (ICME)*, Aug. 2001.
[16] K. A. Peker, R. Cabasson, and A. Divakaran, "Rapid generation of sports video highlights using the MPEG-7 motion activity descriptor," *Proc. SPIE*, vol. 4676, pp. 318–323, Jan. 2002.
[17] N. Babaguchi, Y. Kawai, and T. Kitashi, "Event based indexing of broadcasted sports video by intermodal collaboration," *IEEE Trans. Multimedia*, vol. 4, pp. 68–75, Mar. 2002.

[18] Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for TV baseball programs," in *Proc. ACM Multimedia*, 2000.
[19] K. N. Plataniotis and A. N. Venetsanopoulos, *Color Image Processing and Applications*. Berlin, Germany: Springer-Verlag, 2000, pp. 25–32 and 260–275.
[20] A. Hanjalic, "Shot-boundary detection: Unraveled and resolved?," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 90–105, Feb. 2002.
[21] A. Ekin and A. M. Tekalp, "A framework for analysis and tracking of soccer video," *Proc. SPIE*, Jan. 2002.
[22] G. Millerson, *The Technique of Television Production*, 12th ed. New York: Focal, March 1990.
[23] A. M. Ferman and A. M. Tekalp, "A fuzzy framework for unsupervised video content characterization and shot classification," *J. Electron. Imag.*, vol. 10, no. 4, pp. 917–929, Oct. 2001.
[24] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. New York: Academic, 1999.
[25] V. Kobla, D. DeMenthon, and D. Doermann, "Identifying sports videos using replay, text, and camera motion features," *Proc. SPIE*, Jan. 2000.
[26] H. Pan, P. van Beek, and M. I. Sezan, "Detection of slow-motion replay segments in sports video for highlights generation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 2001.
[27] H. Pan, B. Li, and M. I. Sezan, "Automatic detection of replay segments in broadcast sports programs by detection of logos in scene transitions," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 2002.
[28] Laws of the game and decisions of the international football associations board—2001. [Online]. Available: www.fifa.com.
[29] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, 2nd ed. Albany, NY: Brooks/Cole, 1999, pp. 256–260.
[30] A. M. Ferman and A. M. Tekalp, "Efficient filtering and clustering methods for temporal video segmentation and visual summarization," *J. Vis. Commun. Image. Represent.*, vol. 9, no. 4, pp. 336–351, Dec. 1998.

Ahmet Ekin received the B.S. degree with the highest honors in electrical and electronics engineering from Bogazici University, Istanbul, Turkey, in 1999. He is currently pursuing the Ph.D. degree in electrical and computer engineering at the University of Rochester, Rochester, NY, where he received his M.S. degree in March 2001.

His research interests are real-time multimedia processing, video indexing and retrieval, content-based video summarization, object-based video representations, and applications of state-space models in multimedia processing. He has been working as a Consultant for Eastman Kodak Company since September 1999. During Summer 2001, he was a Summer Intern at AT&T Labs, Middletown, NJ.

Mr. Ekin is a student member of IEEE Signal Processing and Computer Societies.

A. Murat Tekalp (S'80–M'84–SM'91–F'03) received the M.S. and Ph.D. degrees in electrical, computer, and systems engineering from Rensselaer Polytechnic Institute (RPI), Troy, New York, in 1982 and 1984, respectively.

From December 1984 to August 1987, he was with Eastman Kodak Company, Rochester, New York. He joined the Electrical and Computer Engineering Department at the University of Rochester, Rochester, NY, in September 1987, where he is currently a Distinguished Professor. Since June 2001, he is also with Koc University, Istanbul, Turkey. His research interests are in the area of digital image and video processing, including video compression and streaming, video filtering for high-resolution, video segmentation, object tracking, content-based video analysis and summarization, multi-camera surveillance video processing, and protection of digital content. He is the Editor-in-Chief of *Image Communication*. He authored the Prentice Hall book Digital Video Processing (1995). He holds five U.S. patents. His group contributed technology to the ISO/IEC MPEG-4 and MPEG-7 standards. He was an associate editor for *Multidimensional Systems and Signal Processing* (1994–1999). He was an area editor for *Graphical Models and Image Processing* (1995–1998). He was also on the editorial board of *Visual Communication and Image Representation* (1995–1999).

Dr. Tekalp received the NSF Research Initiation Award in 1988, was named as Distinguished Lecturer by IEEE Signal Processing Society in 1998, and was awarded a Fulbright Senior Scholarship in 1999. He has chaired the IEEE Signal Processing Society Technical Committee on Image and Multidimensional Signal Processing (Jan. 1996–Dec. 1997). He has served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING (1990–1992), IEEE TRANSACTIONS ON IMAGE PROCESSING (1994–1996). He was appointed as the Technical Program Chair for the 1991 IEEE Signal Processing Society Workshop on Image and Multidimensional Signal Processing, the Special Sessions Chair for the 1995 IEEE International Conference on Image Processing, the Technical Program Co-Chair for IEEE ICASSP 2000 in Istanbul, Turkey, and the General Chair of IEEE International Conference on Image Processing (ICIP) at Rochester, NY in 2002. He is the founder and first Chairman of the Rochester Chapter of the IEEE Signal Processing Society. He was elected as the Chair of the Rochester Section of IEEE in 1994–1995.

Rajiv Mehrotra is the Business Development Manager in the Entertainment Imaging Division of Kodak, Rochester, NY. His current responsibilities include business opportunities in the area of media asset management and digital right management. He founded Kodak's Media Asset Management R&D program in 1996 and managed it from 1996 to 2001. Prior to joining Kodak, he held faculty positions at the University of South Florida, Tampa, FL, the University of Kentucky, Lexington, KY, and the University of Missouri, St. Louis, MO. He coauthored the book, *The Handbook of Multimedia Information Management* (Prentice-Hall, Englewood Cliffs, NJ, 1997).

Dr. Mehrotra was co-editor of a special issue of IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING ON MULTIMEDIA INFORMATION SYSTEMS (August 1993) and a special issue of *IEEE Computer* (December 1989) on image database management. He is on the editorial boards of *IEEE Multimedia* and *Pattern Recognition Journal* and has served on the program/organizing committee of several international conferences.